Integrated Tools for Mission Operations Teams and Software Agents[1]

Jane T. Malin
NASA Johnson Space Center
2101 NASA Road 1
Houston, TX 77058-3696
281-483-2046
malin@jsc.nasa.gov

Kathy Johnson
University of Texas
Health Science Center
7000 Fannin St., Ste. 600
Houston, TX 77030
713-500-3920
Kathy.A.Johnson@uth.tmc.edu

Arthur Molin
Carroll Thronesbery
S&K Electronics
1016 Hercules
Houston, TX 77058
281-244-5602
thronesbery@jsc.nasa.gov

Debra Schreckenghost
Metrica/Traclabs
1012 Hercules
Houston, TX 77058
281-244-6134
ghost@ieee.org

Abstract—A concept and prototypes for a web-based tool set for mission operations teams are described. This tool set is designed for collecting and distributing information concerning events and issues during missions. Key elements are a console logger, a report generator and a workspace for collecting files, links and actions associated with an issue. The tool set is designed for use both by human team members in a mission control center and by software agents. Software assistants will carry out instructions that automate collection, presentation and distribution of information from diverse sources. One information source can be an autonomous agent controlling a space system or spacecraft. The goal is to integrate software agents with tools that support the work processes of teams, thereby making it easier to automate elements of team tasks and to support team interaction with autonomous agents in space.

TABLE OF CONTENTS

1. INTRODUCTION

In current and future manned space missions, increasing complexity and reduced manpower result in the need for safe, autonomously operating systems in space, and for increased automation and integration among teams on earth. Intelligent software will need to collaborate with and support collaboration among distributed mission operations teams. In the Human Centered Autonomous and Assistant Systems Testbed (HCAAST) project, we are designing and prototyping systems that support such collaboration. Our research and development focuses on two types of agents for supporting space missions:

- Intelligent systems management agents (ISMAs) for autonomous operation in space. ISMAs will provide information about operational and failure events in space, including diagnoses and plans.
- Intelligent Briefing and Response Assistants (IBRAs) for use in space and on earth. Users will customize Briefing and Response Instructions (BRIs) that specify what IBRAs will do. IBRAs will collect, record, analyze and communicate information from ISMAs and other mission sources.

In the work reported in this paper, we focus on Team Work Center, a web-based tool set for use by both the intelligent software agents and mission operations teams. We have begun our work with this focus because of the changes that we have observed in mission operations with the evolution of the International Space Station. This new era has ushered in more complex and demanding multi-team and multi-mission communications, new forms of reporting, new forms of shift handovers and shift reports, and problems with information management. We are focusing on information management tools, because intelligent agents will be valued for the support they can provide in this context. Initially, we are focusing on concepts for supporting mission operations personnel on earth during an event in space. We are working closely with other projects that have the potential for spinning off our prototypes and developing production versions.

As we develop strategies for supporting future human-agent teams, we have been using analyses of cooperative work in the Mission Control Center (MCC) at NASA Johnson Space Center. This information is also used in developing and evaluating prototypes that implement the strategies [1, 2]. For the past ten years we have studied the work of flight controller teams in Space Shuttle and International Space Station operations [3, 4, 5, 6, 7, 8, 9, 10]. We have observed and analyzed teamwork, work processes and products. This work has been used to define prototype web-based tools for console logging and for managing information related to anomalies [11, 12, 13].

These methods of observation and analysis support not only user interface design, but also design of system architecture and software. To discover how to support users performing difficult tasks, it is important to understand user responsibilities and goals, and the context of tasks they are performing. It is also important to understand what they find difficult and how they cope with those difficulties. These methods of analysis and observation have been called user-centered design.

> We do not wish to ask how to improve upon an interface to a program whose function and even implementation has already been decided. We wish to attempt User Centered System Design, to ask what the goals and needs of the users are, what tools they need, what kind of tasks they wish to perform, and what methods they would prefer to use. We would like to start with the users, and work from there [14, p.2].

More recently, these methods have been called contextual inquiry [15], human-centered design, human-centered computing, or human-centered requirements elicitation [16, 17]. Human-centered and contextual methods emphasize understanding the current work setting and the social organization of cooperative teamwork. They look beyond tasks and tools, to analyze how products are used in communication and in ways that at first might appear not to be task-related.

The HCAAST project faces particular challenges for human-centered analysis, because we are prototyping for future teams of human and software agents. It is difficult to find analogs in current work settings for user tasks that do not currently exist and or will be significantly modified in the future. A goal of the HCAAST project is to further develop human centered methods for understanding human-automation teamwork. An additional goal is to enhance these methods to better integrate them into requirements engineering for advanced software development.

This paper provides a progress report, focused on the integrated tool set. We begin with the products of analysis: an explanation of software goals, a description of operational context, and a description of relevant control center activities and tasks. We then describe the support strategy that uses collaborative intelligent software with the integrated tool set. Finally, we describe our first demonstration.

2. Operations Concept and System Goals

Operational Context

The intent of the HCAAST suite of tools is to support manned space operations of the future, as illustrated in Table 1. Intelligent software should support both vehicle and ground operations.

Table 1 – Operations Concept: Context and Players

Context: Ground Rules
- Future missions
- Intelligent software assists both vehicle and ground operations

Vehicle/Crew
- Increased level of crew/vehicle autonomy
- Includes both crew and ISMA software
- Software does the vigilant monitoring
- Crew manages by exception

Multi-Discipline Officers
- Includes both person and IBRA software
- Vigilant monitoring, notify specialists of problems
- High level of data abstraction
- Works shifts in MCC only

Discipline Specialists
- Flight controller functions
- Includes both person and IBRA software
- Intermittent monitoring
- Lower level of data abstraction
- Works in MCC and office
- Responsibilities beyond current mission

Vehicle/Crew—On the vehicle, there is expected to be an increased level of crew and vehicle autonomy beyond current day operations. To simplify the analysis, we are initially concentrating on vehicular regulation processes like Environmental Control and Life Support Systems (ECLSS). Software, including future ISMAs, should perform continuous vigilant monitoring and control, whereas the crew should perform intermittent monitoring and management by exception (take over when there is a problem).

Flight Controllers—On the ground, we anticipate two types of flight controllers: multi-discipline officers and discipline specialists. The multi-discipline officers are expected to monitor operations in the MCC 24 hours a day, seven days a week, responding to any control situation that might occur. Unlike current MCC flight controllers, there will be fewer multi-discipline officers, and they will be monitoring a greater amount of telemetry data. Supported by IBRA agents, they will monitor operations at the level of coordinated, high-level events, rather than the level of individual telemetry readings. Software should also assist them in constructing a good response to those events. One of those responses is to pass on a reasonable report to the discipline specialist.

Discipline Specialists—The discipline specialist of the future is much more like the flight controller of today, an authority on the operations of one aspect of the vehicle. In our initial investigations, we will concentrate on ECLSS operations as a discipline specialty. Like flight controllers of today, they will probably have responsibilities beyond the current mission. However, unlike past MCC operations, in

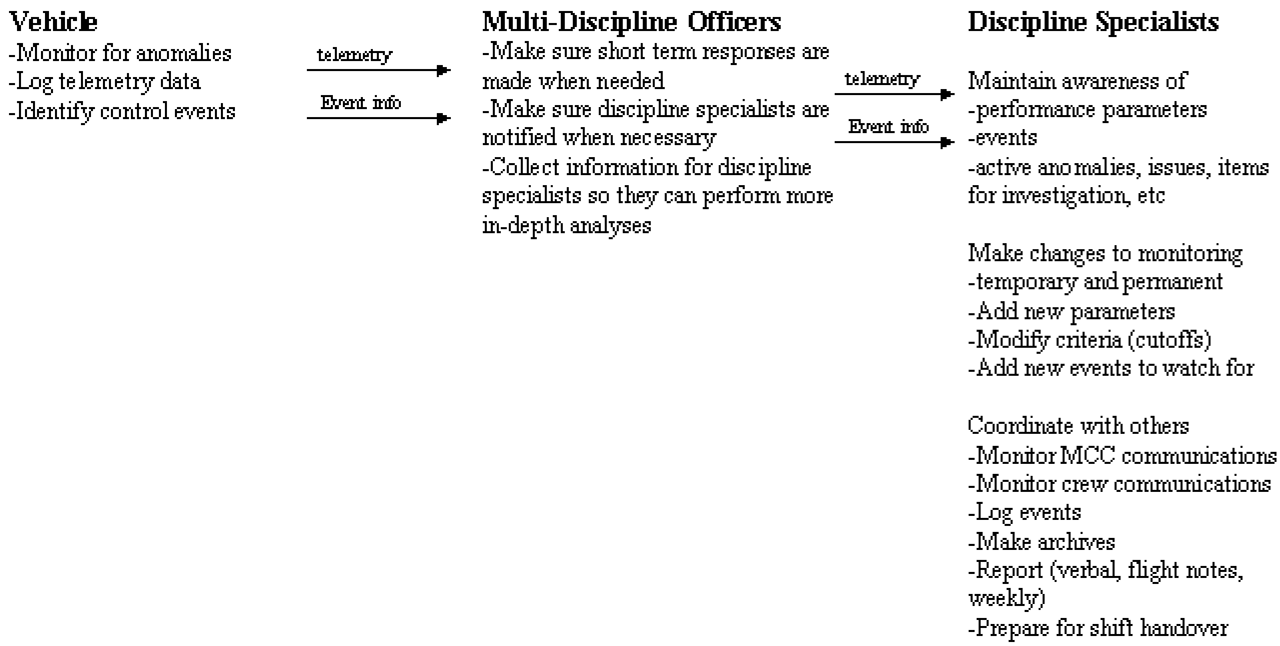


Figure 1 – Functions in Nominal Operations

addition to planning future missions, there may be more than one current mission for the discipline specialist to follow. As a consequence, it will be necessary to relieve the discipline specialists from most of the vigilant telemetry monitoring which characterizes MCC operations of today so that they can concentrate on evaluating anomalies identified by software and multidiscipline officers, reviewing mission events for additional anomalies, and planning future operations. Because of the wide variety of responsibilities, it seems likely that many of the discipline specialist activities will be performed in their offices rather than in the control center.

For this operations concept to be judged safe, it will be necessary to demonstrate that multi-discipline officers, with the aid of intelligent software, can respond appropriately to anomalies and notify discipline specialists in a timely and effective manner. MCC already has some experience with operations similar to the multi-discipline officer from the early days of the International Space Station [4]. Before there was a crew onboard the Station, in the nighttime hours when no commanding telemetry was being sent from the ground, Station Duty Officers (SDOs) observed the telemetry for evidence of nominal quiescent operations. If an anomaly occurred, they followed pre-specified instructions (Anomaly Response Instructions, or ARIs) prepared by the discipline specialists. Those instructions included safing the system, diagnosing the anomaly, collecting data to be used by the specialists, and possibly calling the specialists to report. The discipline specialists revised the ARIs as they gained experience with them, and updated them as knowledge of the Space Station changed.

Major Tasks

Figure 1 identifies the major functions to be performed during nominal operations at each of three centers of activity. The first center is at the vehicle. Most of the ECLSS activities at the vehicle should be automated. Monitoring for anomalies, logging data, and identifying important control events (e.g., changes in operation modes of the life support equipment) can be handled by automation. In fact, intelligent systems currently perform these functions while controlling experimental life support systems in ground tests at Johnson Space Center.

The second center of activity is organized under the responsibilities of the multi-discipline officers. As indicated earlier, they vigilantly monitor ongoing operations, make short-term responses when needed, and collect information for the discipline specialists so that they can review mission data more efficiently.

The third center of activity is organized around the responsibilities of the discipline specialists. With the help of automation, they will maintain mission cognizance (to respond quickly and effectively when an anomaly arises), and examine mission data for evidence of hidden anomalies or the need to adjust operating parameters. To perform these tasks properly, they will need to coordinate with others, make log entries and prepare reports. HCAAST automation should help with these coordination tasks, as well as with the analyses and mission awareness tasks.

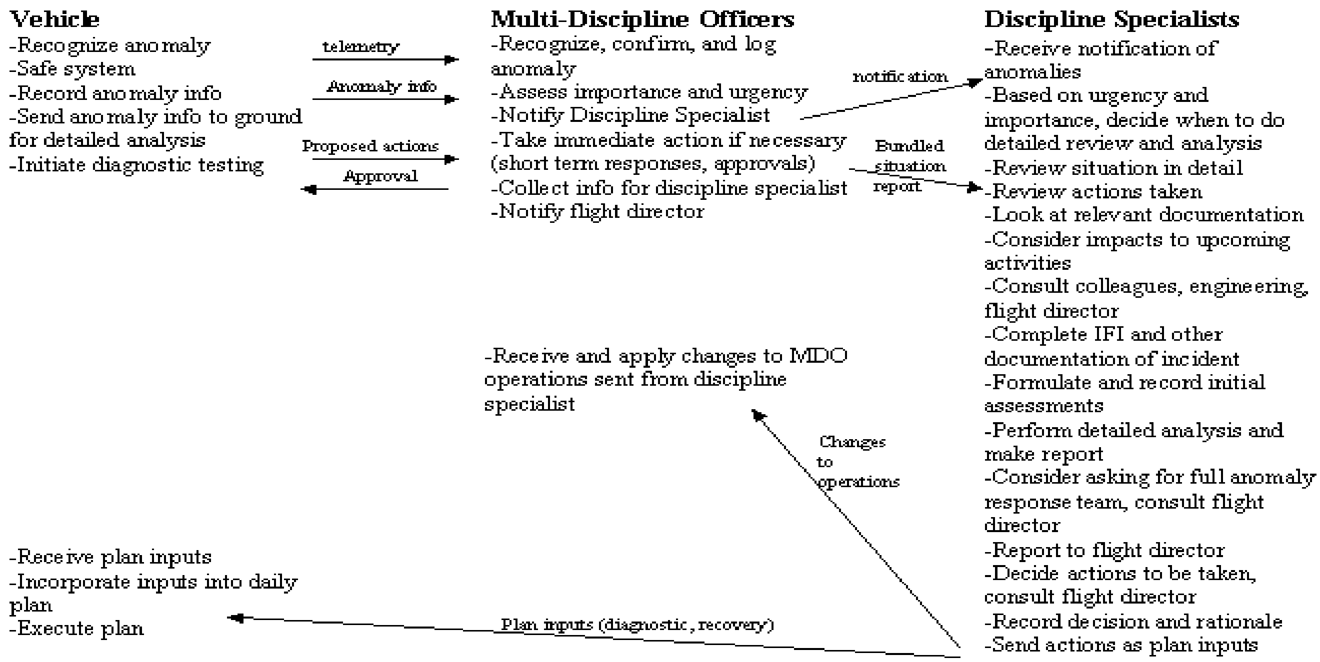


Figure 2 – Functions in Anomaly Response

Figure 2 identifies similar functions at each of these locations during an anomaly response. To simplify our initial efforts, we are focusing on support for the discipline specialist. Nominal and anomaly response operations are somewhat different and involve some differences in information that must be exchanged. However, it is important that we understand both types of activities because any tools we develop will need to support both.

In our current work, we are focused on supporting control center personnel as they produce and use information products during shifts and handovers. Figure 3 shows types of information sources and information handling. A primary task is to get a stable record of selected material from streams of rapidly changing distributed information. The chronological console log or diary is the primary means used to stabilize this information. The information is further refined and combined with diverse interim and semi-permanent information sources, to produce summary reports, action lists and formal request packages. These products are used to orient other team members as they prepare to come on shift and take over responsibilities. Other participants also use and review these products from various remote locations as they work cooperatively on issues and plans, and they may respond as needed with new interim information sources.

Currently, handling these diverse sources of information is challenging. It is difficult to attend simultaneously to all the volatile information sources and keep up, while preparing for the next relevant operation and participating in work on plans and issues. Selecting and recording information from rapidly changing sources into console logs can be tedious and repetitious. Word processors and spreadsheets, multiple software tools and voluminous information make it difficult to find needed items and move them among tools and team members without errors or loss of critical information. It is cumbersome to manage information access for various groups. It is time-consuming to conform to work processes that are not well supported by the tools. Software agents can assist in information handling and help address the challenges of these tasks.

System Goals – Supporting Humans Doing Difficult Tasks

Probably the most important goal of Human-Centered Computing is to ensure that systems support cooperating teams with the difficult parts of their tasks. Therefore our first step has been to determine the set of high-level tasks that humans are or will be performing. We term these high level tasks "system goals." We start with goals derived from the task lists in Figures 1 and 2, in combination with our study of flight controllers during previous and related software projects [11, 12, 13, 18, 19, 20], interviews with and observations of flight controllers [3, 7, 8], and study of future operations concepts written by mission operations personnel [21]. Because we intend to support tasks in a future operations environment that is somewhat different from current mission operations, we have also looked at some situations that are analogous [22] to such future environments. More recently, we have studied the intermittent monitoring of data from ECLSS laboratory tests that are ongoing at Johnson Space Center. This project

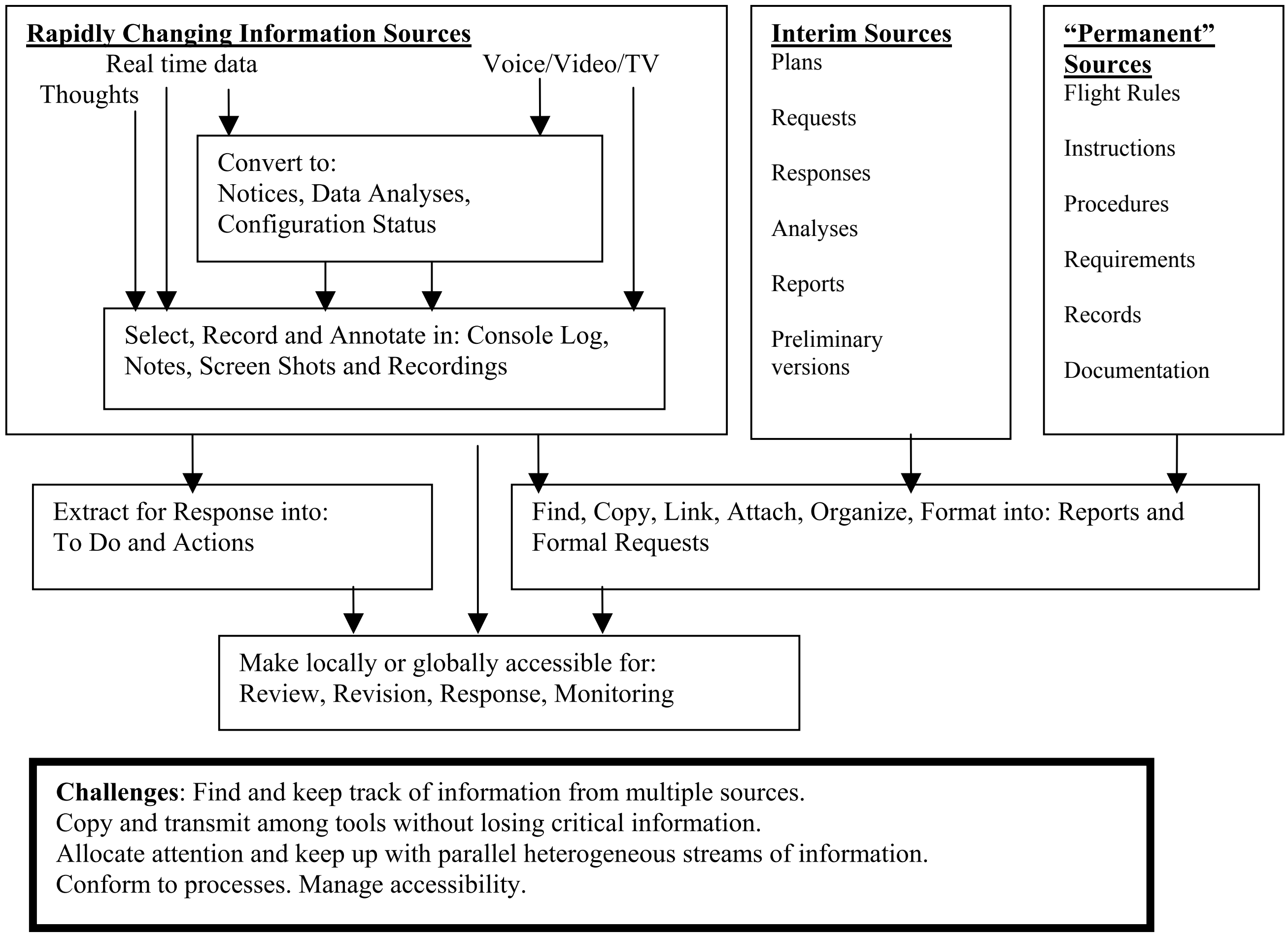


Figure 3 – Information Handling on Control Center Shifts

gives us additional insight into intermittent monitoring and into interactions with semi-autonomous intelligent control systems similar to what we expect to see aboard space vehicles of the future. Additional information about how to fulfill these goals has come not only from the above sources but also from artifacts (handwritten console logs, routine reports, anomaly reports) and software tools currently used. By looking at currently used tools and artifacts, we can better understand the human tasks and we can see ways in which the tools and artifacts have been modified to fit those tasks. Figure 3 shows one of the products of our recent analyses that focused on the information flow during a shift in the Mission Control Center.

From this analysis combined with others, we have determined a list of software goals with rationale that we will now describe.

Support intermittent monitoring and evaluation—Discipline specialists will have a wide range of responsibilities, some of which extend beyond the current mission. These specialists need to spend time efficiently, which means they cannot vigilantly watch telemetry flows of hundreds of sensor readings. In order to work efficiently, they will need to be able periodically check on the mission status and health, and evaluate mission data for hidden anomalies and for the need to adjust operational parameters. They will decide if near-term planned activities need to be adjusted, take any needed actions, and then get on to their next task. Another important part of support for intermittent monitoring is to provide the discipline specialists with the knowledge that they will be notified in a timely manner of any anomalies that arise [22].

Support mission cognizance—Shuttle flight controllers have indicated that mission cognizance is extremely important. Knowing the current status (health) of the equipment, the current state of plan execution, and the issues and anomalies that are currently active enables them to respond quickly and effectively when a new anomaly arises. By the same token, the intermittent monitoring of ECLSS experimental equipment on the ground requires a certain level of situational awareness in order to understand the data obtained from the intermittent checks. The challenge is for the software to capture this information effectively and to present it in a way that makes it easy to see the big picture view of the mission without restricting access to mission data [6].

Provide efficient descriptions of apparently nominal events—Discipline specialists will need to look at apparently nominal events and decide whether to look at them in more detail for indications of hidden anomalies and for indications that operating parameters should be adjusted. This means that an overview of the event should be presented and that details for any part of the event should be readily accessible by the user on demand [12].

Provide timely notification of anomalies—For the users to have any confidence in monitoring on an intermittent basis, they need to know that they will be notified on a timely basis when anomalies arise. The intelligent system challenge will be to ensure that important anomalies requiring urgent responses are not missed and that false alarms are kept to a minimum. The human centered computing challenge is to tailor the content and format of the notification to the users role (discipline specialist on duty vs. off-duty), location (MCC, office, home), and personal preferences (email, pager, phone call with synthesized speech, etc.), and with the urgency of the message [3].

Provide anomaly description—Once the discipline specialist has been notified of an anomaly and begins to analyze it, s/he needs to be able to gain a quick understanding of what the anomaly is all about. That high level understanding will guide a more detailed analysis to evaluate candidate hypotheses about the situation, so that the discipline specialist can formulate an appropriate anomaly response. The software needs to assist in this entire process.

Direct attention and keep up to date—There is a tremendous amount of information available to flight controllers and much of their expertise is knowing what is important to attend to at any given point in time. We anticipate the amount of telemetry data for which a given flight controller is responsible for monitoring will increase in the future. We also anticipate more interleaved tasking on top of mission data monitoring (e.g., more activity planning). Consequently, software tools of the future will need to assist the flight controllers in knowing where to direct their attention.

Support review and handover—Even if intermittent monitoring is supported well, it seems unlikely that one person can be responsible for a given discipline for the duration of a long-term mission. That means that a handover of flight controller responsibilities needs to be supported so that the incoming flight controller can respond just as effectively as the outgoing one. Many of the same tools that assist the flight controller in maintaining mission cognizance should be of assistance here. However, it seems wise to identify this as a separate goal because there will probably be additional software requirements to support handing over to a new flight controller [7].

Support anomaly analysis and response: group coordination—Many anomalies are recognized and the flight controller can respond to them without performing extensive analysis. However, some anomalies require some in-depth analysis. In these cases, an anomaly response team is formed to work the problem, bringing a wider range of expertise as well a more manpower to bear. Participants in the process can be dispersed geographical and temporally. Software should support them in sharing the results of their work with one another as well as planning and organizing the anomaly response process itself [11].

Support other team interactions—There are additional coordination tasks, as implied by Figures 2 and 3. They involve informing others of actions, making reports available, monitoring communications among people of other flight disciplines, asking for specific information, asking for agreements from specific people, etc. These tasks can be very taxing in an integrated MCC environment where representatives of all flight disciplines are gathered in a central location at the same time. It is more taxing when several control centers and office personnel are involved. We anticipate that these tasks will be even more challenging if the future involves discipline specialists doing much of their work from their offices. Software is needed to assist these tasks [9].

Copy and transmit among tools without losing critical information—Some current flight controllers are experiencing a specific set of coordination problems that need to be addressed soon. When they have prepared reports, data, etc., that are needed by other people, they do not have an easy way to notify everyone concerned that the information is ready and then to make it available to them. The problem is exacerbated by the fact that many consumers can use the same reports for specific purposes. Most people using the information in a report are not interested in that entire report. As a result of this lack of notification and lack of access, flight controllers can be distracted from their main tasks to answer phone calls requesting information and to place phone calls to notify people that information is available. They need assistance in notifying all interested parties of the existence of the information and then putting that information in a place accessible and recognizable by the people who need it. Most importantly, software tools need to be integrated to relieve flight controllers of the added tasks of copying and distributing information once it has been generated.

Manage accessibility—Some information is sensitive, especially that related to personal medical information. On the other hand, nearly all information needs to be made accessible to someone other than the flight controller who is currently on console. Software tools for flight controllers need to be integrated so that the right people have ready access and so that access to work in progress is withheld until it is ready [11].

Help locate relevant documentation—Flight controllers have a number of rich documentation sources for guiding them in performing anomaly analysis, performing diagnostics and determining appropriate responses. Software should assist them in identifying the sources of documentation relevant to a given anomaly or event summary, and in finding the relevant location within each of these sources for the task at hand.

Keep track of information from multiple sources—During the course of analyzing an anomaly, the flight controller may need to access a number of documentation sources (engineering drawings, previous mission anomaly reports, flight rules, fault analysis procedures). In addition to knowing which of these are relevant to a given anomaly, the flight controller has the difficult task of integrating the knowledge gained from these sources.

Conform to processes—Finally, there are a number of established processes that help mission control activities flow in an orderly fashion, ensuring safety and thorough analysis of mission data and coordination among MCC personnel. While software tools may not always need to rigidly enforce these practices, they should conform to them and help flight controllers conform to them.

3. COLLABORATIVE INTELLIGENT SOFTWARE

An important objective of this project is the investigation of how humans and software agents can work together to accomplish space operations more effectively than humans working alone. Specifically, we have looked at two types of software agents: onboard ISMA agents that automate formerly manual tasks and customizable IBRA agents that assist humans in performing manual tasks.

ISMA system managers handle routine operation of onboard systems. They can provide descriptions and analyses of both nominal operations and anomalies in space systems, as well as information on the state (beliefs, plans, activities) of ISMAs themselves. They can produce high-level information about operations, plans, and fault detection, diagnosis and recovery. ISMAs will be further discussed in Section 5.

IBRA agents can help users maintain situational awareness and prepare for anomaly response. IBRAs can take information from ISMAs and other sources and assist crew and ground personnel in the following ways:

- Convert volatile data and information to **notices** and **data analyses**, collect and organize data associated with an event.
- Automate and assist getting volatile data and information into shift **logs and notes**, producing filtered notes, updating and searching for selected entries.
- Automate and assist pulling together and linking data, information, log entries and notes into **reports.**
- Automate and assist distributed review and revision and update processes for **reports** and **official paperwork requests**.
- Automate and assist selecting, collecting/linking, organizing, archiving and accessing distributed stable data in shareable issue-focused **workspaces.**

Figure 4 shows how these agents can interact with crew and ground.

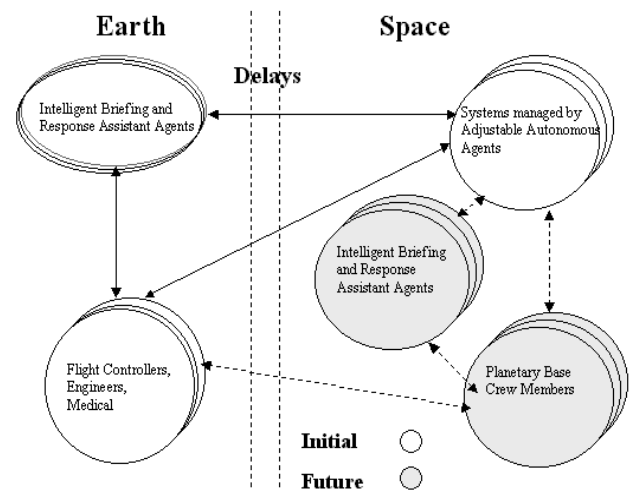


Figure 4 – Agent Interactions

The mission operations environment is collaborative in nature. The crew interacts with multi-discipline officers who interact with each other as well as the discipline specialists who also interact with each other. Each has large numbers of tasks to perform as efficiently as possible. Intelligent software can support such collaborations through the automation and facilitation of monitoring and communications tasks. At the simplest level of intelligence and automation, software agents can automatically detect events and generate reports and notices. The previous section suggested many areas in which software can support mission operations tasks. Our first efforts focus on tools to support the discipline specialists.

Let us consider a malfunction in a device responsible for removing carbon dioxide (CO_2) on a spacecraft. If there is no automation involved, a person must notice this problem – either crew or a multi-discipline officer. In either case, one must notify the other so that the situation can be corrected. In addition, the environmental and medical discipline specialists must be notified. More detailed information must then pass between all of the relevant people who come together as a team to solve the problem. An ISMA on the vehicle could automatically detect the problem with the CO_2 removal device and generate a notice of the problem and any initial steps taken automatically to resolve the problem. The multi-discipline officer or IBRA assistant can use this notification to create a report on the problem. A notice about this report can be posted as on a general status page.

Taking the example a bit further, the notice could also be sent by pager, email, voice mail, or other method to those who need to know about the incident, such as the environmental and medical discipline specialists. The first step for determining who needs to know could be by user subscription to a notification service. Adding intelligence and knowledge to the system, the system itself could infer, through the use of roles and knowledge of systems, the appropriate people to notify. Here, the software agent becomes a sort of glue joining the other team members together through intelligent communication of relevant information. This agent can even keep track of whether a person has accessed the information and escalate its method of contact or contact person depending on the urgency of the notification. The fact that this happens automatically means that no single person can be a bottleneck for long and that no person needs to vigilantly monitor the situation. Ultimately, information flows to the people who need it in a very efficient manner, leaving more time for the task of working any issues raised.

An ISMA agent is not likely to be able to fully generate the information that the multi-discipline officer would include in a report upon receiving notification of a problem. However, an IBRA could gather the relevant information such as device specifications, flight rules, and data values from the notice or telemetry that would be needed for the multi-discipline officer to create the report. IBRAs can be customized to fit a particular team member's style and expectations. For example, an administrator might want just the briefest notice, an engineer may want the entire report immediately, and a flight controller may prefer to just receive the brief notice and later pull the entire report. With just a little more intelligence, the agent itself may infer the style of interaction a team member prefers based on past history. Providing the right amount of information at the right time is a key to working effectively in teams.

4. Tool Set and Products

A key element of our approach to the needs and challenges of the mission operations environment is a Team Work Center, a suite of linked tools for information handling that flight controllers and software agent can both use. This suite is being designed with the following features:

- Web-based for global access and database-based for tool-independent access and search
- Data can be easily transmitted or linked between tools without extra data entry effort or error.
- Explicit content links between tools support finding things and keeping track of their locations.
- Agents can be incrementally developed to automate information-handling tasks.
- Agents can be specified and maintained more easily by users if they use the same infrastructure.
- Learning approaches that reuse techniques of users can be more easily supported.

The tools in the Team Work Center will include:

- Electronic Console Logger - to create a database of log entries, which supports review of large log files, automated logging, and generation of reports and specialized logs
- Workspace Manager - to collect and share items related to an issue or anomaly or work topic in one accessible workspace, with capability to handle files, links, actions, logs, and paperwork
- Report Maker - to create report formats and collect information from multiple sources (log entries, data, data analyses, notices, actions, procedures, links to workspaces and references) into formatted reports. The information in the report can also be edited, rearranged, annotated and updated. The report can be associated with a workspace. Example reports could be notices, handover reports, and reports about anomalies that include information from ISMAs.
- Notifier - for managing notification of team members on or off console
- Instructions and Procedures - for specifying automation and team processes and procedures, including Briefing and Response Instructions (BRIs).

The concept for BRIs is based on the anomaly response instructions (ARIs) that were discussed in Section 2. A BRI contains the instructions that IBRA will follow when a triggering message or data value is received. A BRI can specify who needs to be notified in response to the anomaly, and what data needs to be included in the notification. If data analysis is needed or a detailed report needs to be prepared, the BRI will specify what data needs to be contained in the report. The BRI can also specify what is to be entered into the console log, and may specify that a workspace be created or that a report be entered into an existing workspace.

We have made significant progress in prototyping two of the tools in this Team Work Center, the Electronic Console Logger and the Workspace Manager.

Electronic Console Logger. The web-based logger tool can fulfill the same roles as the console diary kept by flight controllers, a time-stamped chronicle of important mission and events and progress on tasks [13]. Figure 5 shows a

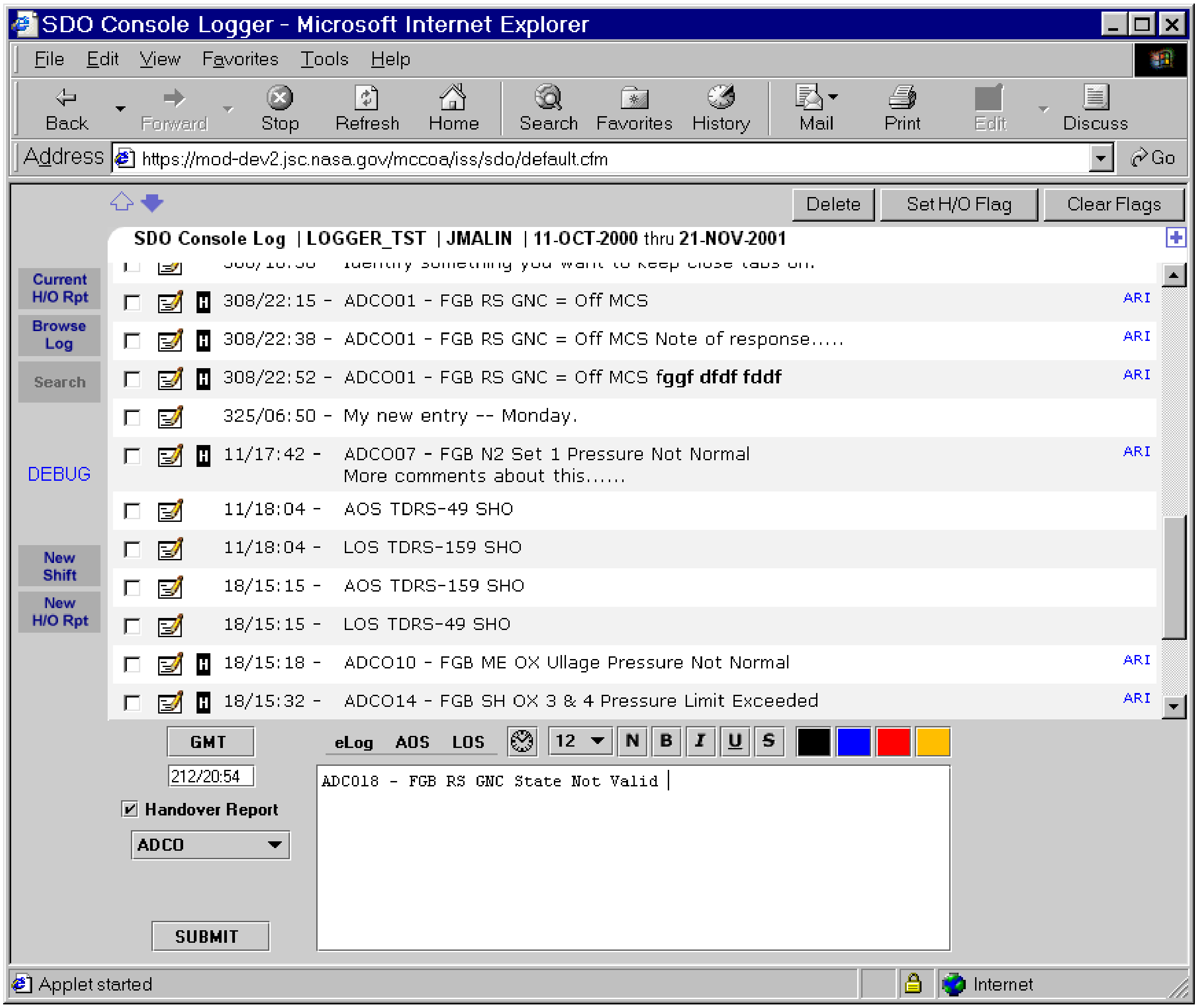


Figure 5 – Electronic Console Logger

screen from the current prototype. Since it is web-based, logs are readable by incoming flight controllers before they arrive at MCC. It stores entries in a searchable database, enabling flight controllers to compare current mission events to past events. The database enables sorting and filtering entries for review, and selecting them to include in reports and specialized logs. The database also enables copying, linking and transmitting log entries into other specialized logs or reports. An entry can be as simple as a selected piece of data or notice from telemetry that has been seen on a data display. An IBRA agent could enter such items in the log, helping a flight controller keep up with rapidly changing data. An entry may be annotated with commentary and may be updated after it is entered. The entry may refer to related materials that are not in the log, but are linked to the entry. An entry can record the beginning, progress, or end of an operation, with commentary on performance and issues. An entry can note a voice communication or conversation, and its progress. An entry may provide rationale and reminders on ongoing work on plans or issues.

Workspace Manager (WorkIT). The web-based workspace manager tool is based on a prototype to support anomaly response teams [11]. Information on a topic can be collected from multiple sources, organized and tracked. Information that is relevant to an issue be easily located, and information access can be managed. Discipline or project workspaces can be created to provide integrated access to diverse material that is used or produced as a team works on a task. The team may be working on an issue or anomaly, a plan, a procedure or other design. The workspace can be temporary, or it can be archived.

Figure 6 shows a portal-style design for a specific workspace, with the latest files and links, actions and specialized issue logs. This workspace portal view

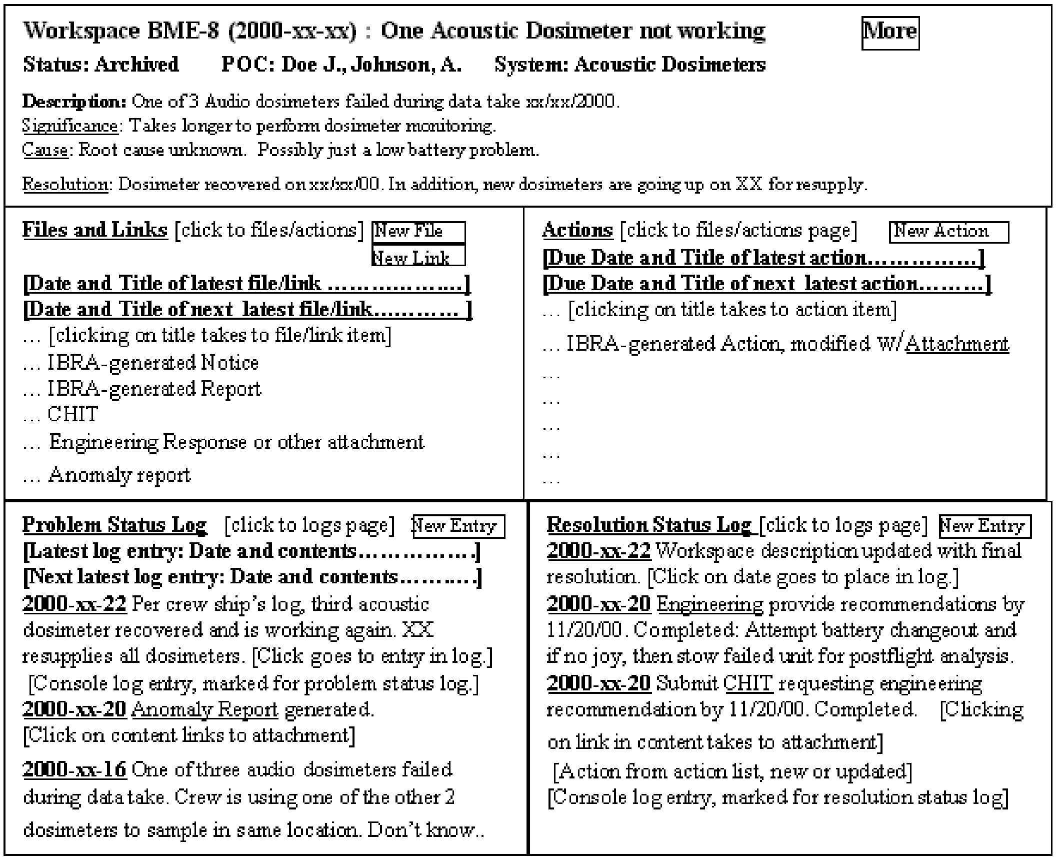


Figure 6 – Concept for Portal View of a Workspace

illustrates some of the close interactions envisioned among the tools, as content from various sources is viewed together in a new workspace location and links lead back to information in its original locations in other tools. The concept also illustrates some software agent responsibilities such as the "IBRA-generated Action" in the Action pane.

Figure 7 shows the current prototype of WorkIT in the contents view of a particular workspace, called "HIPAA". Items in the workspace can be assigned to varying levels of access. Global items would be visible to international team members, Internal items would be visible to anyone on the Johnson Space Center team, and Private items would be visible only to discipline team members working the issue. This allows for draft items to be constructed and worked before making them more globally available.

5. First Demonstration

A demonstration was prepared to show the concepts of the Team Work Center and the IBRA agent, interacting with an ISMA type of agent handling a problem with a life support subsystem. The Team Work Center demonstration shows how a flight controller on the ground could investigate an anomaly by using reports on the actions of ISMA and system events.

The ISMA that we have developed is autonomous control software that can be adjusted to accommodate human intervention [23]. To achieve this high level of autonomy, the ISMA architecture is comprised of multiple, interacting software components, in the Three Tier Control Architecture developed for autonomous robot control [24].

Planner—predicts crew and software activities needed to achieve control objectives. The *Planner* models both humans and software as agent resources available to achieve mission goals. It supports closed loop control by (1) monitoring plans as they execute, (2) detecting when task plans fail to achieve their goals, and (3) re-planning in response to the failure to achieve goals.

Control—consists of two components, the *Sequencer* that selects and orders procedures to implement planned activities and the *Skill Manager* that implements procedure steps as closed loop control. The Sequencer chooses procedures reactively, based on the current state of environment. It allocates procedure steps to specific skill managers. Skills are connected to a simulated system. Skills are activated to issue commands to control instrumentation modeled in this simulation. Events are activated to monitor sensor readings in the simulation in response to control.

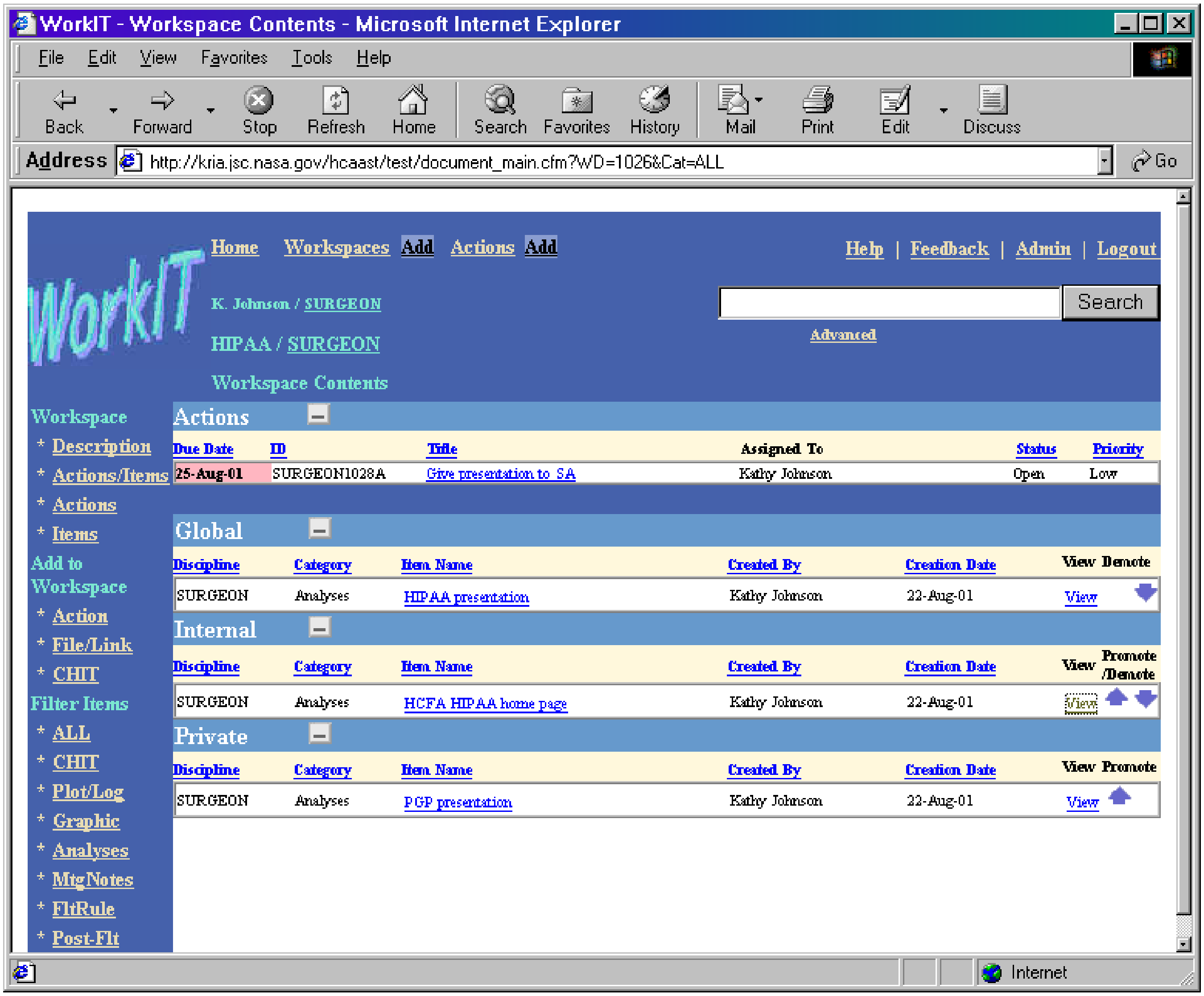


Figure 7 – Contents View of a Workspace

Fault Detection, Isolation, and Recovery (FDIR)—the model-based reasoner, Livingstone [25], models systems as components connected by data propagation paths. Livingstone monitors commands sent to systems and data from these systems. It can diagnose failures by predicting the consequences of observed commands, based on its models, and comparing these predictions to the observed consequences. Livingstone can also reason over its models of behavior to determine reconfiguration actions in response to anomalies.

In the demonstration, an ISMA agent controls a simulation of a crew life support subsystem for CO_2 removal. This simulation was developed using the CONFIG advanced simulation environment [26]. CONFIG provides an object-oriented modeling language that supports both qualitative and quantitative modeling of components. It also provides for modeling of control operations. These models can be exercised using a discrete event simulation engine. CONFIG provides a graphical user interface for building and exercising models. Figure 8 shows the ISMA architecture integrated with the CONFIG simulation.

Air Revitalization System

We tested and demonstrated our concept of human-agents teams using a simulation of a subsystem of the crew Air Revitalization System (ARS). Figure 9 shows a high level view of the crew air revitalization process. The ARS is a space-based crew support system that regenerates the atmosphere in a crew habitat. Its purpose is to remove CO_2 produced by crew metabolism from the air and convert it to oxygen (O_2). There are three subsystems in the ARS:

- Variable Configuration Carbon dioxide Removal (VCCR) System: molecular sieve technology with beds that are cycled to extract CO_2 molecules from the air, for storage in a storage tank.

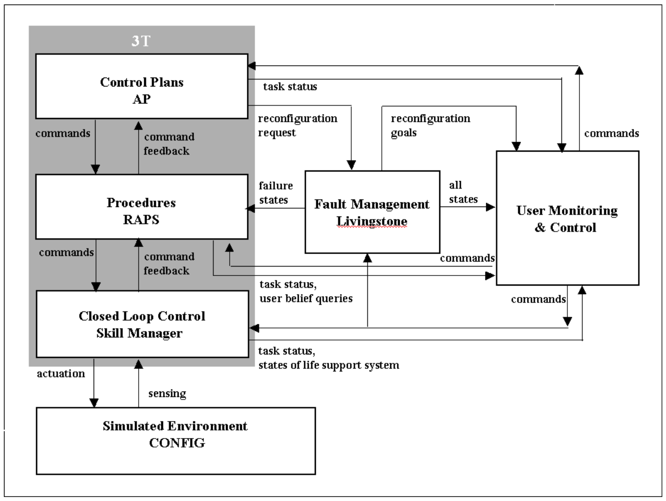


Figure 8 – ISMA and the Life Support Simulation

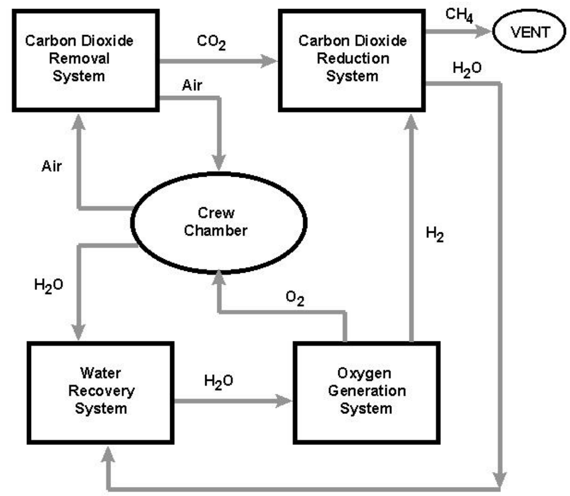


Figure 9 – Air Revitalization Process

- Carbon dioxide Reduction System (CRS): high temperature process that combines hydrogen (H_2) with CO_2 from the storage tank, to produce methane (CH_4) and water (H_2O).
- Oxygen Generation System (OGS): electrolytic process for converting water produced by the CRS into the constituent O_2 and H_2 molecules. O_2 is transferred into the crew habitat and H_2 is transferred to the CRS.

Figure 10 shows the graphical user interface for the simulation model for the VCCR. B3 and B4 are used in alternation to adsorb and desorb CO_2 from cabin air. B1 and B2 are used for humidity control. CO2-BUFFER is the gas storage tank.

Scenario

The scenario used to illustrate human-software agent interaction in mission operations is the detection of an anomaly in the VCCR. This anomaly is first detected when a lower than expected concentration of CO_2 is observed in the gas storage tank. The following sequence of events describes how the agents in the system react to this event.

- ISMA/Control detects a lower than expected concentration of CO_2 in the gas collected during the last half cycle from bed 3. Gas with CO_2 concentration below 85% is not usable by other systems in the ARS. ISMA/Control immediately stops the VCCR from removing CO_2 and safes the system.

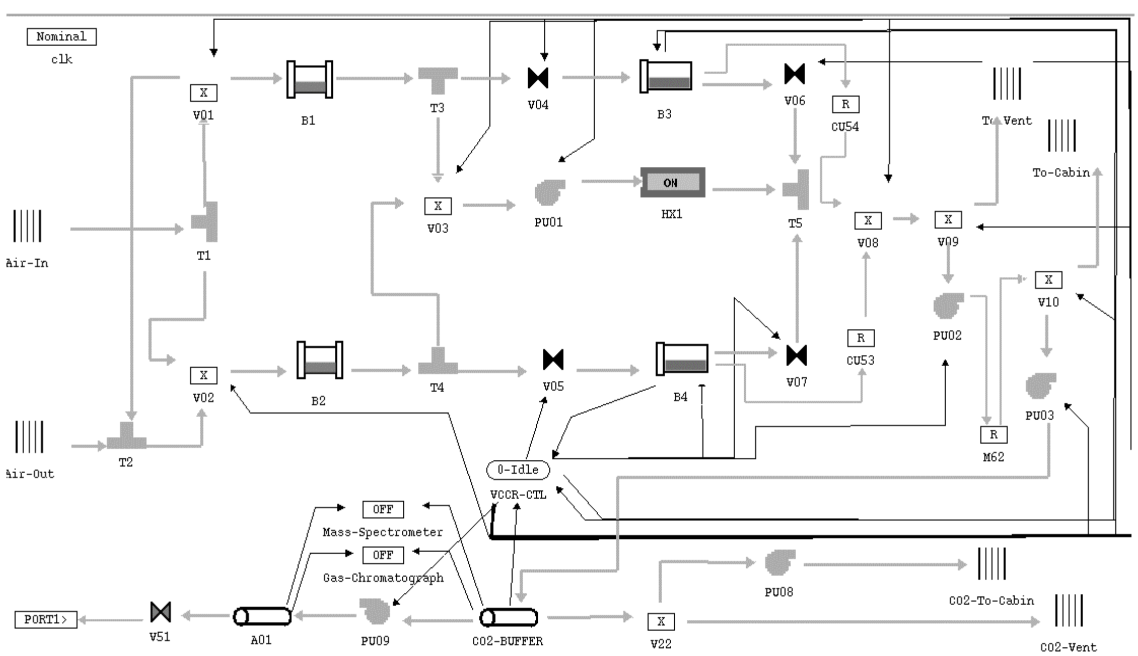


Figure 10 – Model of Variable Configuration CO_2 Removal System

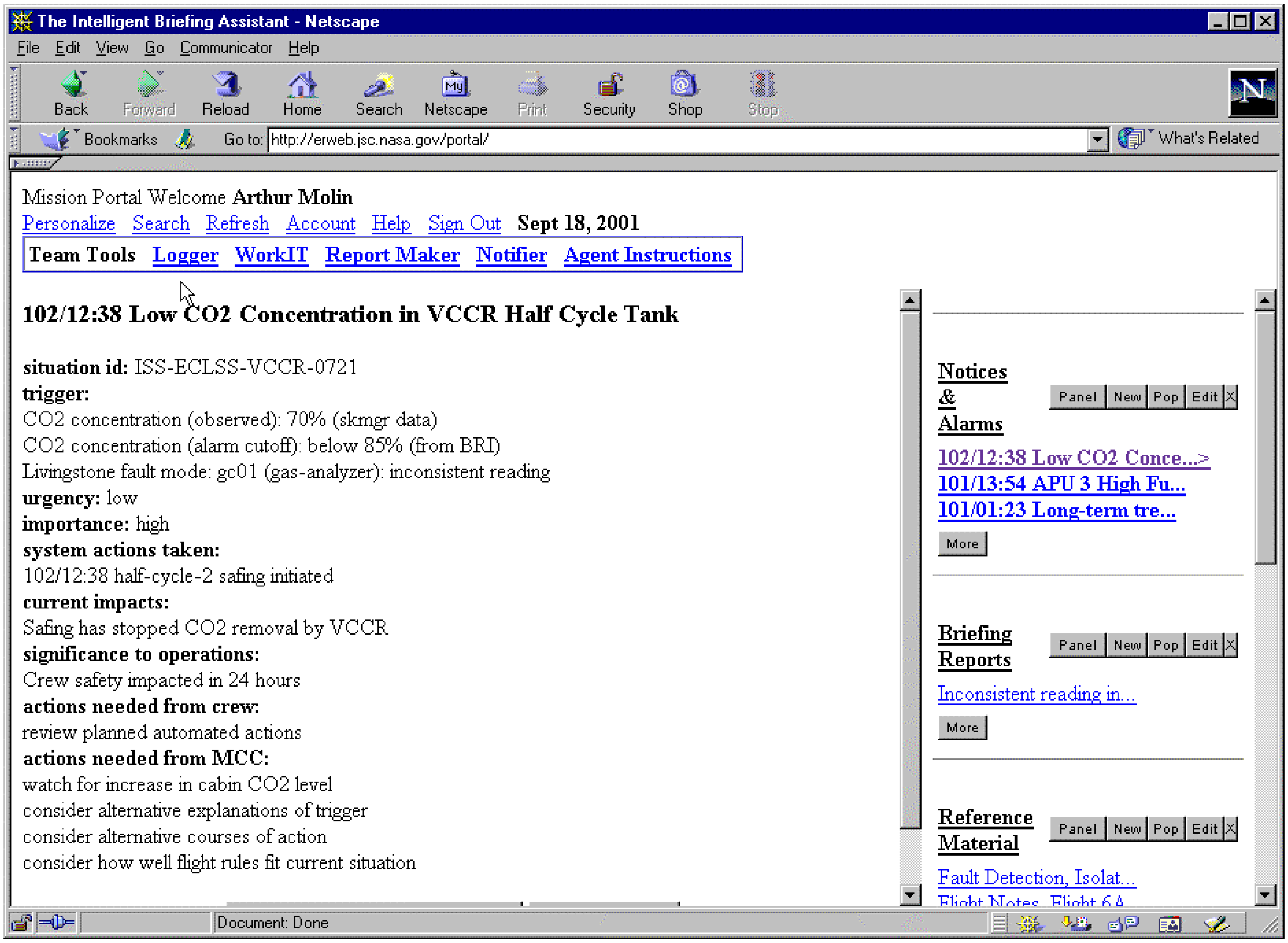


Figure 11 – Notification Received from IBRA

- ISMA issues a local notice that the VCCR has been reconfigured to the SAFE configuration in response to the low CO_2 concentration from bed 3.

- ISMA/FDIR issues a message indicating that the gas chromatograph (GC) that took the reading is in an anomalous state (inconsistent_reading), but it does not have enough information to isolate the cause. Both the crew and flight controllers are notified that FDIR has issued a message on the GC being used by the VCCR.

- The GC message triggers the generation of a notice and a report by IBRA that summarizes the state and status of the VCCR when the anomaly is detected, and the control actions taken in response to anomaly detection. This notice is shown in Figure 11. If the CO_2 removal equipment is off-line for an extended period of time, the flight controllers may need to replan exercise periods and increase monitoring of cabin air quality.

- ISMA/Planner reacts to the inconsistent_reading state of the GC by scheduling a diagnostic test that helps determine the cause of the anomaly. The planned test is to take a sample of a known CO_2 source using the same GC. If the reading matches the known value, then the instrument is healthy and the low concentration indicates a leak in bed 3. If the reading does not match the known value, then the instrumentation is faulty and switching to a backup gas analyzer should fix the problem. IBRA includes the information on the scheduled test in the report (not shown).

At this point, the anomaly detection phase of failure response is complete and autonomous system is preparing for the fault isolation phase. ISMA agent components would work together to diagnose and recover. This activity is not included in the demonstration, which covers only the first phase of failure response, detection and reporting of the anomaly.

Figure 11 shows the Team Work Center after a notice of the anomaly has been received and displayed. In this demonstration, an IBRA reacts to a triggering message from ISMA and generates a notice and a report. The trigger for IBRA to develop the notice and the report is CO_2 concentration below 85%. The IBRA helps the flight controllers maintain situational awareness during an evolving situation such as the one in this scenario. IBRA can direct the attention of the flight controller to important information, while filtering out irrelevant information. In the scenario, IBRA directs the attention of the flight controller to the anomaly, and provides information that will be needed for response.

The notice is a description of the anomaly that occurred, along with ISMA actions, impacts, and actions that will likely be needed by the crew and flight controller team. If the user needs more information to determine the correct course of action, a detailed report is also available for review. The report can provide additional information for the flight controllers responding to the anomaly:

- Detailed records of telemetry values preceding the anomaly, presented graphically
- Records of other important events that preceded (or followed) the anomaly
- Links to relevant areas of the electronic log
- Actions planned and taken by the automated system

In addition, the report can also provide areas for annotation; the IBRA agents and/or the flight directors can add information to the report as it becomes available.

The instructions that IBRA follows to develop the notice and report are specified in a BRI. The notice and report can include data about states of the system and ISMA, as well as predefined information about expected impacts and actions needed. In this demonstration, all the system data came via messages from ISMA. The information in the notice in Figure 11 was encoded in the Extensible Markup Language (XML). One advantage of XML is that the report data is separated from information about how the report is to be presented. Flexible displays of the report can be separately implemented. The presentation format for the notice in Figure 11 is simple, since at this stage in our project we have focused on information content more than on presentation.

6. Conclusions and Future Work

The human-centered analysis approach has helped us define complementary roles for two types of agents in future space operations, by directing our attention to the context of control center teamwork and its challenges. We have identified the IBRA as an intermediary that manages information from the ISMA as well as other sources. This analysis has clarified the importance of the IBRA, a customizable assistant type of agent that can be incrementally developed and changed by the users. This analysis has also led us to define an information management work context for IBRA agents themselves, the Team Work Center, which they share with the operations team. The use of the web and databases can support sophisticated information reuse, and help to solve various problems that arise from copying and transmitting information from place to place. The information management tool set may help partition the functions of the agents and make their activities more controllable and understandable.

The design concept we have developed may well be applicable in a number of settings where distributed and asynchronous teams work together to manage operations, and could benefit from automation. Such teams are at work in many domains including space, medicine, the military and industrial manufacturing and processing.

There are a number of elements of the design that are not complete. Next, we plan to work on a prototype of a Report Maker tool and on a tool integration architecture to support the Team Work Center concept. We will also focus on agent models for IBRA agents, and on approaches for providing users the capability to specify and update agent instructions in the context of the Team Work Center. We will continue to use ISMA control of simulated life support systems for demonstrating the evolving design. Next, both operations personnel and IBRAs will collaborate on developing and updating reports and other products, and will handle a series of events in a group of life support subsystems.

Acknowledgment

This work is funded in the Human Centered Computing area of the Intelligent Systems Program managed by NASA. The authors want to thank Mike Shafto for his constant support of this interdisciplinary work. We wish to thank Steve Gonzalez for his support at Johnson Space Center. We wish to thank members of the HCAAST team for their excellent work on this project: Land Fleming, Grace Lei, David Overland, Bryan Porcher, Dan Smith and Shashi Srinivasa. We also want to thank individuals have contributed substantially to the underlying foundations of this work: David D. Woods, Susan Torney, John O'Neill, Roxana Wales, Chuck Shultz and LaDessa Hicks.

References

[1] C. G. Thronesbery and J. Malin. *Field Guide for Designing Interaction with Intelligent Systems.* NASA Technical Memorandum, NASA TM-1998-208470, 1998.

[2] Malin, J.T.; & C.G. Thronesbery. "Integrating Human Factors Knowledge into Design of Intelligent Systems: Lessons Learned." Poster presented at Human Systems Conference, Houston, TX, 2001.

[3] R. Chow, K. Christoffersen and D. D. Woods, “A Model of Communication in Support of Distributed Anomaly Response and Replanning,” *Proceedings of XIVth Triennial Congress of the International Ergonomics Association and of the Human Factors and Ergonomics Society 44th Annual Meeting, **1**, 34-37*, 2000.

[4] R. Chow, *A model-based analysis of flight control logs as artifacts of communication and coordination.* (ERGO-CSEL Report 00-TR-03). The Ohio State University, Institute of Ergonomics/Cognitive Systems Engineering Laboratory, 2000.

[5] J. Malin, D. Schreckenghost, D. D. Woods, S. Potter, L. J. Johannesen, M. Holloway and K. Forbus. *Making Intelligent Systems Team Players.* (NASA Technical Report 104738). Johnson Space Center, Houston TX, 1991.

[6] E. S. Patterson, *Coordination across shift boundaries in space shuttle mission control.* (CSEL Report 1997-TR-01). The Ohio State University, Cognitive Systems Engineering Laboratory, 1997.

[7] E. S. Patterson and D. D. Woods, “Shift changes, updates, and the on-call model in space shuttle mission control,” *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting,* (*243-247*), 1997.

[8] J. Watts-Perotti and D. D. Woods, *A cognitive analysis of functionally distributed anomaly response in space shuttle mission control.* (CSEL Report 1997-TR-02). The Ohio State University, Cognitive Systems Engineering Laboratory, 1997.

[9] J. Watts, D. D. Woods and E. S. Patterson, “Functionally Distributed Coordination During Anomaly Response in Space Shuttle Mission Control,” *Human Interaction with Complex Systems '96*, Dayton, OH, 1996.

[10] J. Watts, D. D. Woods, J. Corban, E. S. Patterson, R. Kerr and L. Hicks, “Voice Loops as Cooperative Aids in Space Shuttle Mission Control,” *CSCW '96 Proceedings*, Boston, MA., *48-56*, 1996.

[11] J. T. Malin, L. Hicks, D. Overland, C. G. Thronesbery, K. Christoffersen and R. Chow, "Creating a Team Archive during Fast-Paced Anomaly Response Activities in Space Shuttle Missions." (NASA Technical Publication). Houston, TX: NASA-JSC, 2002.

[12] C. Thronesbery, K. Christoffersen and J. Malin, “Situation-Oriented Displays of Space Shuttle Data,” *Proceedings of the Human Factors and Ergonomics Society 43d Annual Meeting*, September 27- October 1, 1999.

[13] S. Torney and J. Ortiz, “Operations Assistants for the Manned Space Program,” *2001 IEEE Aerospace Conference Proceedings*, March 10-17, 2001.

[14] D. Norman and S. Draper, *User Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum, 1986.

[15] H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann: San Francisco, 1997.

[16] S. Viller and I. Sommerville, “Coherence: an Approach to Representing Ethnographic Analysis in System Designs,” *Human-Computer Interaction* 14 (1 & 2): 9-41, 1999.

[17] C. Potts and I. Hsi, “Abstraction and Context in Requirements Engineering: Toward a Synthesis,” *Annals of Software Engineering*, **9**, *1-39*, 1997.

[18] R. Chow, K. Christoffersen, D. Woods, E. Patterson, J. Malin, C. Thronesbery and D. Schreckenghost, “Why Mission Control Works: Supporting Cognition and Collaboration in Distributed, Dynamic Worlds.” Panel Overview in the *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting*. Houston, TX, 1999.

[19] J. T. Malin, C. Thronesbery and S. A. Land. "End-Effector Monitoring System: An Illustrated Case of Operational Prototyping," *Proceedings of the Seventh Annual Workshop on Space Operations and Applications Research (SOAR '93)* (NASA Conference Publication). Houston, TX: NASA-Johnson Space Center, 1993.

[20] K. A. Johnson and M. C. Shek, “User and Task Analysis of the Flight Surgeon Console at the Mission Control Center of the NASA Johnson Space Center,” *Report on Summer Faculty Fellowship at NASA Johnson Space Center,* 2000.

[21] *Operations Concept Definition for the Human Exploration of Mars* (DV-00-014) 2d ed. Houston: Johnson Space Center, 2000.

[22] C. G. Thronesbery and D. L. Schreckenghost, “Human Interaction Challenges for Intelligent Environmental Control Software.” *Proceedings of the 28th International Conference on Environmental Systems*. Danvers, MA, 1998.

[23] D. Schreckenghost, J. Malin, C. Thronesbery, G. Watts, and L. Fleming, “Adjustable Control Autonomy for Anomaly Response in Space-based Life Support Systems,” *Workshop on Autonomy, Delegation and Control* at International Joint Conference on Artificial Intelligence, August, 2001.

[24] P. Bonasso, J. Firby, E. Gat, D. Kortenkamp, D.Miller, and M. Slack,. Experiences with an Architecture for Intelligent, Reactive Agents. *Journal of Experimental Theory of Artificial Intelligence*. 9: 237-256, 1997.

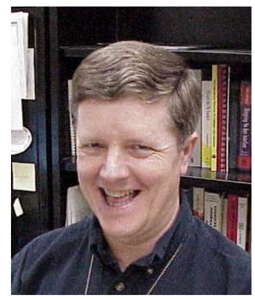
[25] B. C. Williams and P. P. Nayak, "A Model-based Approach to Reactive Self-configuring Systems," *Proceedings of the 13th National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA, (971-978), 1996.

[26] J. T. Malin, L. D. Fleming, and D. R. Throop, "Hybrid Modeling for Scenario-Based Evaluation of Failure Effects in Advanced Hardware-Software Designs," *Model-Based Validation of Intelligence*, Technical Report SS-01-04, AAAI Press, Menlo Park, CA, 2001.

Jane T. Malin *is Technical Assistant and Computer Engineer in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center, where she leads intelligent systems research and development. She received a Ph.D. in Experimental Psychology, specializing on cognitive psychology, from the University of Michigan in 1973. She has led development of the CONFIG simulation tool for evaluating intelligent software for operation of space systems. She has led research on intelligent user interface and expert systems for real-time monitoring and fault management of space systems. She is principal investigator in the project, Human Centered Autonomous and Assistant Systems Testbed.*

Kathy A. Johnson *is Assistant Professor of Health Informatics at the University of Texas Health Science Center at Houston. She also serves as Medical Informatics Lead at NASA Johnson Space Center. She received a doctorate in Computer Science with studies in Artificial Intelligence from The Ohio State University in 1993. Her specific interests are in the areas of intelligent tutoring systems, decision support systems, and interface design. She currently teaches courses on databases and knowledge modeling in health informatics. A major project has been the design and building of a multimedia system for gait analysis including both tutoring functionality as well as providing support for writing intelligent reports. Her most recent project concerns studying and improving the flight surgeon and biomedical engineer console environment.*

Arthur Molin *is a program manager working for S&K Electronics at NASA Johnson Space Center. He has developed a number of artificial intelligence systems, including an on-board procedure assistant for the Space Shuttle and a system to re-write COBOL code to work around the Y2K bug. He has a B.A. from Northwestern University and a M.S. in computer science from the University of California.*



Carroll Thronesbery *has been developing intelligent systems and human-computer interaction software since receiving his Ph.D. in cognitive psychology in 1981. His projects at NASA Johnson Space Center include intelligent systems to help flight controllers monitor flight data, collaborative software to support anomaly response teams, and human interaction with intelligent life support controls software. His research interests include human-intelligent system interaction to support the control and monitoring of engineered systems (e.g., life support systems, Shuttle hardware sub-systems).*

Debra Schreckenghost *received a Masters Degree in Electrical Engineering at Rice University in 1982. After graduating, she worked at NASA Johnson Space Center as a Shuttle flight controller. For the last 10 years she has applied advanced software technology to aid humans in performing space operations. She was technical lead for the development of an adjustable autonomy control system to manage product gas transfer during the Lunar/Mars Life Support Technology Program Phase III test. Recent areas of research include architectures with adjustable levels of control autonomy, mixed initiative planning, intelligent user interfaces, and visualization of situated action.*